

## CLAIMS

1. A method for decoding video data, said method comprising:

writing a start code starting at a byte in a middle portion of a data word in a memory;

writing a starting address associated with the byte in a table; and

fetching data from the memory starting from the byte.

2. The method of claim 1, wherein the start code is associated with a slice.

3. The method of claim 1, wherein the data word comprises at least 16 bytes.

4. The method of claim 1, further comprising:

writing another start code to another byte in a middle portion of another data word in the memory;

writing another address associated with the another byte in the table; and

wherein fetching data from the memory starting from the byte further comprises:

fetching data from the memory starting from the byte and ending with a byte preceding the another byte.

5. The method of claim 4, further comprising looking up the address in the table.

6. The method of claim 5, further comprising looking up the another address in the table.

7. A system for decoding video data, said system comprising:

a memory comprising a plurality of data words, for storing a start code starting at a byte in a middle portion of a particular one of the data words;

a table for storing a starting address associated with the byte; and

a direct memory access module for providing data from the memory starting from the starting address.

8. The system of claim 7, further comprising:

a video transport processor for writing the start code starting at a byte in a middle portion of the particular data word in the memory.

9. The system of claim 7, wherein the start code is associated with a slice.

10. The system of claim 7, wherein the data word comprises at least 16 bytes.

11. The system of claim 7, wherein the video transport processor writes another start code to another byte in a middle portion of another data word in the memory and wherein the table stores another address associated with the another byte in the table and wherein the direct memory access module fetches data from the memory starting from the byte and ending with a byte preceding the another byte.

12. The system of claim 11, further comprising:

a master processor for looking up the address in the table.

13. The system of claim 12, wherein the master processor looks up the another address in the table.

14. The system of claim 7, wherein the direct memory access module further comprises:

a buffer comprising a plurality of data words for storing the video data from the starting address;

a first masking register for discarding a portion of a first data structure that precedes the starting address, the first masking register comprises a plurality of bytes corresponding to byte positions; and

a state machine for loading the first masking register with a pattern wherein each byte of the plurality of bytes in the first mask register that corresponds to a byte position that is less than the four least significant bits of the starting address are loaded with a first value, and wherein each byte of the plurality of bytes in the first mask register that corresponds to a byte position that is equal or greater than the four least significant bits of the starting address are loaded with a second value.

15. The direct memory access module of claim 14, wherein the first value is a hexadecimal 0 and wherein the second value is a hexadecimal F, and further comprising:

an arithmetic logic unit for performing a logical AND operation between a first one of the plurality of data words in the buffer and the first masking register.

16. A direct memory access module for providing video data from a starting address to an ending address, said direct memory access module comprising:

a buffer comprising a plurality of data words for storing the video data from the starting address and the ending address;

a first masking register for discarding a portion of a first data structure that precedes the starting address, the first masking register comprises a plurality of bytes corresponding to byte positions; and

a state machine for loading the first masking register with a pattern wherein each byte of the plurality of bytes in the first mask register that corresponds to a byte position that is less than the four least significant bits of the starting address are loaded with a first value, and wherein each byte of the plurality of bytes in the first mask register that corresponds to a byte position that is equal or greater than the four least significant bits of the starting address are loaded with a second value.

17. The direct memory access module of claim 14, wherein the first value is a hexadecimal 0 and wherein the second value is a hexadecimal F, and further comprising:

an arithmetic logic unit for performing a logical AND operation between a first one of the plurality of data words in the buffer and the first masking register.

18. The direct memory access module of claim 14, further comprising:

a second masking register for discarding a portion of a second data structure that follows the ending address, the second masking register comprising a plurality of bytes corresponding to byte positions; and

a state machine for loading the second masking register with a pattern wherein each byte of the plurality of bytes in the second mask register that corresponds to a byte position that is less than or equal to the four least significant bits of the ending address are loaded with the second value, and wherein each byte of the plurality of bytes in the second mask register that corresponds to a byte position that is greater than the four least significant bits of the ending address are loaded with the first value.